

Towards Optimally Efficient Field Estimation with Threshold-Based Pruning in Real Robotic Sensor Networks

Amanda Prorok, Christopher M. Ciani and Alcherio Martinoli

Abstract—The efficiency of distributed sensor networks depends on an optimal trade-off between the usage of resources and data quality. The work in this paper addresses the problem of optimizing this trade-off in a self-configured distributed robotic sensor network, with respect to a user-defined objective function. We investigate a quadtree network topology and implement a fully distributed threshold-based field estimation algorithm. Simulations with field data as well as real robot experiments are performed, validating our distributed control strategy and evaluating the threshold-based formula for real world scenarios. We propose a theoretical analysis that predicts the system's behavior in real world case studies. The experiments and this prediction show very good correspondence, enabling the accurate employment of the objective function, optimizing the trade-off based on user needs.

I. INTRODUCTION

Since the beginnings of research on sensor networks in the 1970s, the monitoring of environments and habitats has become one of its major application fields [3]. Technological advances in embedded systems, such as the development of reliable wireless communication, and miniaturization and improved efficiency of microcontrollers and sensors have answered key needs, and encouraged an increasing deployment of wireless sensor networks as a main tool to monitor spaces [7]. Still, one of the challenges presented with the deployment of sensor networks is the accurate estimation of fields with unpredictable environmental phenomena, while simultaneously addressing the critical issues of resource usage such as local memory, communication and processing constraints.

With networks often consisting of a considerable number of sensor nodes, the necessity of limiting energy consumption as well as bandwidth requirements increases. Research in the domain of ad hoc wireless routing has produced a range of algorithms which propose solutions for these problems. Improved routing algorithms have been developed which aim to accomplish in-network load balancing and an increased system lifetime, employing techniques that are mostly based on system information such as remaining energy levels and routing capacities. For instance, the GAF (*Geographical Adaptive Fidelity*) scheme of Xu et al. [23] superposes a constant grid onto the network area. Nodes use their geographic location to determine grid membership and transit between idle and active states, ensuring that one node per grid will remain active to route packets. The approach described by

Chen et al. in the *Span* scheme [2] is similar, with the difference that nodes do not need localization. Decisions on whether to power down radio communication are made locally, on the basis of a coordinator-election rule which preserves network routing capacities. In [12], Heinzelman et al. describe LEACH (*Low-Energy Adaptive Clustering Hierarchy*), a clustering-based routing protocol. It employs randomized rotation of local base-stations (clusterheads), with decisions to become a clusterhead depending on remaining node energies.

In wireless sensor networks, a reduction of resource consumption due to routing and sensing tasks may also be driven by field data. The density or activity of the sensor nodes can be adapted to sensing application needs by eliminating redundant nodes and thus reducing energy usage. Intanagonwiwat et al. [13] describe directed diffusion, a data-centric, application-specific approach which aims to minimize the communication distance between sensor nodes and data sinks. This approach is complemented by research done by Zhao et al. [24], [15]. These works present collaborative signal processing techniques for distributed sensor networks, also aiming to minimize resource usage by activating nodes only on a when-needed basis by considering a maximum utility of predictions on sensor measurements. The methodology is developed especially for tracking applications, where communication within the network is based on a neighbor-to-neighbor protocol.

In our work, we address the problem of designing distributed sensor networks for surveillance and monitoring. It is clear from [14] that self-configuration is a necessary element for effective as well as efficient performance of such networks. The proposed design paradigm suggests hierarchical topologies, following a top-down control and bottom-up reconfiguration principle. Here, we build upon this design rule, implementing a distributed, multi-layer tree-based routing algorithm and combining it with a threshold-based clustering strategy which is adaptive to the state of the field being estimated. Our algorithm leans on established field estimation methods described in [19] and [22]. The approach is similar to the one described by Arici et al. in [1], which describes an adaptive sensing method also based on a tree-like, hierarchical network structure. Their method exploits the fact that a manual deployment of sensors may offer more information than necessary (over time and space) to reconstruct an accurate field estimate. They propose a self-configuration algorithm which will put nodes into passive mode when their measurements become 'predictable'. Here, also motivated by previous research in the domain

Amanda Prorok, Chris Ciani and Alcherio Martinoli are with the Distributed Intelligent Systems and Algorithms Laboratory of the Ecole Polytechnique Fédérale de Lausanne. This work was partially sponsored by the Swiss National Science Foundation (Contract Nr. PP002-116913). `firstname.lastname@epfl.ch`

of distributed sensor node controllers as presented in [6], we develop a fully distributed node controller that is easily implemented on resource constrained and noisy hardware, which aims to optimize system performance by finding a trade-off between use of resources and data quality. In contrast to the methods described in [2], [12], [23], we base our clustering strategy on field data, rather than on system information. Also, our resulting data aggregation method follows a multi-layer bottom-up principle, which enables global abstraction of the target field, different from the local collaborative processing methods of [24], [15]. Lastly, in contrast to [19] and [22] we focus on the whole system rather than only on communication and routing activities, and we verify our approach by demonstrating it on real hardware and by comparing the performance to theoretical predictions (see also III-A).

II. CASE STUDY: ESTIMATING AN ACOUSTIC FIELD WITH A ROBOTIC SENSOR NETWORK

First developed with the Distributed Sensor Networks (DSN) program in 1978, the demonstrative target problem for distributed sensor networks was acoustic tracking [3], with numerous studies published in this domain since then. In this study, we attempt to estimate an acoustic field generated by a static sound source in a closed environment with evenly spaced robotic sensor nodes. Nevertheless, our proposed methodology can be applied generally, to any type of field estimation task. Ultimately, we wish to obtain a unified estimation of the field values by extracting this information from the network data sink. Throughout this work, we discuss performance by means of i) mean-squared-error (MSE) between the actual field values and the estimated values, and ii) number of active nodes. Without any loss of generality, we assume that the number of active nodes within the network is proportional to the consumption of system energy, since the power of transmission is the same on all robotic nodes.

A. Distributed Network Organization

As suggested in the theoretical work of [22], we superpose a quadtree (shown in Fig. 1) on the robotic sensor network. Especially when computing spatial problems typical in computer aided design and geo-data applications [11], the quadtree data structure has proven an efficient and powerful tool [10] [20]. An early work in [9] shows how an active quadtree network facilitates image representation and analysis. Also, a recent study in [8] shows how a quadtree can be utilized for in-network data querying in a fully distributed wireless network. Yet, despite the quadtree being a widely known structure, to the best of our knowledge, it has not yet been exploited as a network topology.

Here, although our controllers and models are general to any hierarchical topology, we showcase our study on a quadtree based network with each robotic node within our sensor field representing a leaf node in the tree structure. The robots are distributed on a regular grid in a square arena. In a network of a total n nodes, assuming that the robots are

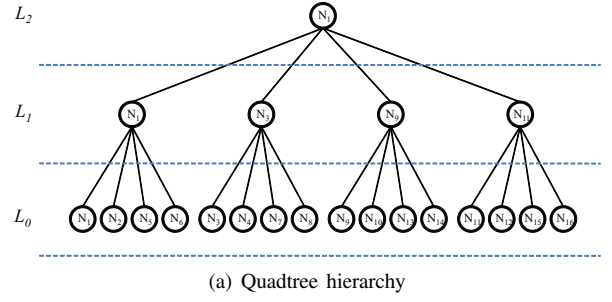


Fig. 1. A 16-node quadtree structure. The quadtree hierarchy is decomposed into 3 hierarchy levels. A node will participate in either of the 3 subsets: $\{L_0\}$, $\{L_0, L_1\}$ or $\{L_0, L_1, L_2\}$.

aware of their location, each one allocates itself to one of n sensing cells in the decomposed space. We thus obtain a robotic sensor network ordered by the intrinsic hierarchy of the quadtree. Adapted and implemented in a fully distributed sensor network, this hierarchy can be explored in terms of i) communication channels and ii) fine-tuning the spatial resolution of the sensor network. Whereas exploring i) is relatively straightforward as we can directly exploit the quadtree hierarchy, there are many approaches to ii)—our chosen approach will be discussed later in Section II-B.2.

On a global level, the quadtree structure depends only on the number of nodes (implicitly a power of 4), and can be constructed in a distributed manner, assuming that all nodes know their location. As is evident in Fig. 1, a single node may have multiple roles within the network, depending on the status of the network. Thus, we create the notion of layers L_i . In a network of 4^K nodes, we have $K + 1$ layers (L_0, \dots, L_K), and a node's current role in the network is defined by its current processing layer $L_{current}$. Every node N_i has a maximum layer $L_{k_{max}}$ with $N_i \in L_{k_{max}}$ such that there is no $k > k_{max}$ with $N_i \in L_k$. Also, any node N_i in $L_k, k > 0$ is a clusterhead, with four descending nodes in L_{k-1} as its cluster children (including itself). As we are here not interested in node failures, we don't go into the details of an eventual clusterhead rotation or election strategy.

The group of robotic nodes uses wireless communication as a means of inter-node organization. There are two classes of messages being used within the network: control messages and data messages (measurements). The messages typically contain the following elements: *measurement/control data*, i and k , with i the id of the sender node N_i and L_k its current processing layer. Control messages are sent top-down through the network structure, and measurement messages bottom-up. Nodes throughout the network or within the communication range of the transmitting node may receive messages at all times and asynchronously from various senders. A clusterhead will only accept measurement data from nodes belonging to its cluster, and following the top-down control principle, a node will only accept control messages from its clusterhead. Fig. 2 illustrates the communication protocol.

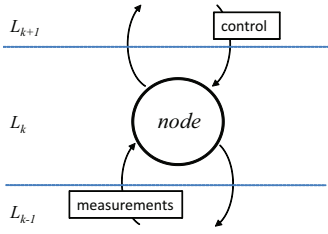


Fig. 2. The node is currently processing data in layer L_k . Measurement messages are sent bottom-up and control messages are sent top-down the quadtree structure.

B. Control of the Robotic Node

We elaborate two control variants: first, a *naive sensing* strategy (NS), and second, an improved *threshold-based sensing* strategy (TBS). With NS, the nodes are in one of three possible states, whereas with TBS, the nodes are in one of four possible states. The controller is simple and distributed, homogeneous on all nodes.

1) *State Machine*: The controller can be represented by a simple state-machine, and is depicted in Fig. 3. Initially, a node is in the *sample* state. Each time a node takes a measurement, it will transition to the *process* state. If the node is a leaf node (its processing layer is $L_{current} = L_{k_{max}}$ at all times) it will transition directly to the *broadcast* state, send its measurement and then return to the *sample* state. If the node is a clusterhead, it will increment its processing layer $L_{current}$ once it has received (and aggregated) the data from all the nodes in its cluster, and will enter the *broadcast* state if it has reached its maximal layer $L_{k_{max}}$. Otherwise, it will re-enter the *sample* state. Finally, upon sending the (collected) measurement data in the *broadcast* state, the clusterhead will return to the *sample* state.

In a further step, we develop the controller for TBS, with the goal of optimizing the use of resources by reducing the number of messages sent and measurements taken. The aim is to *prune* certain node-clusters off the quadtree by putting the nodes in those clusters to sleep. A clusterhead will then replace measurement values of all its descendant nodes with its own. A fourth state is added to the NS controller, and is illustrated by the dashed line in Fig. 3. If a node has received a relevant pruning control message, it will be absorbed by the *idle* state. In this work, we do not consider the reactivation of idle nodes through a tree-branching procedure.

2) *Threshold-Based Pruning Algorithm*: In TBS, a clusterhead makes the decision to prune or not prune its child nodes. Thus, we implemented a threshold-based pruning algorithm, which builds on the theoretical formula proposed in [19]. Assuming that the field is anisotropic, the chosen approach is to prune sensor-node clusters which are sampling values in isotropic subparts of the field. The resulting field estimator will display a higher sensing resolution along the boundaries of the anisotropic field and lower resolution in the isotropic subparts. This principle is illustrated by the

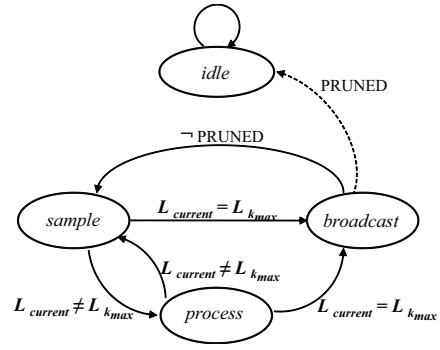


Fig. 3. Schematic illustration of two variant state-machines implemented for the quadtree structure. (a) NS (without dashed line): A node samples acoustic events. Measurement data from cluster nodes is received and processed. When the cluster data is complete, a node will broadcast the collected data. (b) TBS (with dashed line): A node which is shut down is absorbed by the idle state.

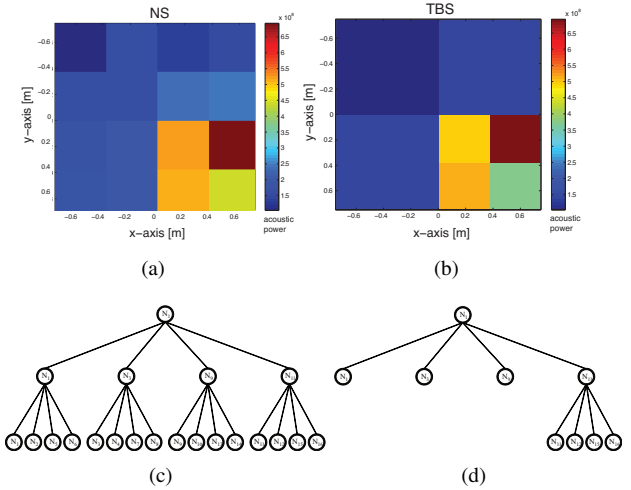


Fig. 4. The graphs show the calculated power of an acoustic event at a given moment. Each of the 16 cells is occupied by one robotic sensor node. An acoustic source is located in the bottom left corner of the arena. (a) A snapshot of the true field values (b) The data sent out of the network by the top-level node after completion of the pruning algorithm

example in Fig. 4. Fig. 4 (a) and (c) show a fully active (un-pruned) quadtree and the values transmitted by the full network, whereas Fig. 4 (b) and (d) show a pruned quadtree and the values transmitted by the remaining active nodes.

From [19] we have:

$$\hat{f}_n = \underset{f(\theta), \theta \in \Theta_n}{\operatorname{argmin}} R(f(\theta), x) + 2s^2 p(n) |\theta| \quad (1)$$

where s^2 is the signal noise variance and $p(n)$ a monotonically increasing function of the total number of nodes. The finite set Θ_n includes all possible pruning variations (partitions) of a quadtree with n nodes, and θ is one particular partition. Then, for the set of partitions Θ_n , the algorithm will seek the optimal partition θ which minimizes the cost of the resulting field estimator, \hat{f}_n . This cost is comprised of two terms. The first term $R(f(\theta), x)$ is the approximation error resulting from the pruned clusters in the partitions. The

error is calculated as in

$$R(f(\theta), x) = \sum_{i=1}^n (f_i(\theta) - x_i)^2$$

where $f_i(\theta)$ is the estimated value for a node N_i in a particular partition θ and x_i is the true field value. The aim of the second term in (1), $2s^2p(n)|\theta|$, is to penalize increasing complexity, where the factor $|\theta|$ is the number of not pruned nodes in the partition. In [18], $p(n) = 2/3 \log n$ and s^2 is homogeneous on all sensor nodes.

We can solve equation (1) in a distributed manner by using the bottom-up messaging protocol mentioned in Section II-A. The work in [18] confirms that both terms of the estimator are additive functions, thus the error and the penalty cost of a subsquare can be calculated by each corresponding clusterhead independently. Then, following our messaging protocol, a clusterhead in the quadtree hierarchy will receive from its 4 child nodes (three child nodes and itself) the field estimate which minimizes the estimation cost as given by the formula.

In order to implement the field estimation technique in our distributed network, we propose a threshold-based pruning algorithm. At layer L_0 , there is no propagated error from lower levels, the cost $\hat{f}_i(\theta_{L_1})$ at a clusterhead N_i is thus equal to

$$\hat{f}_i(\theta_{L_1}) = \begin{cases} 8s^2p & \text{if not pruning} \\ R(f_i(\theta_{L_1}), x) + 2s^2p & \text{if pruning} \end{cases}$$

The algorithm will seek the minimal cost $\min\{\hat{f}_i(\theta_{L_1})\}$, therefore the threshold on the approximation error $R(f_i(\theta_{L_1}), x)$ for layer L_1 is

$$T_1(s, p) = 6s^2p$$

In other words, if the approximation error $R(f_i(\theta_{L_1}), x) < T_1(s, p)$, the cluster will be pruned. For layers L_k with $k > 1$, the estimator takes into account the propagated errors and complexity penalizers from lower level layers, with

$$\hat{f}_i(\theta_{L_k}) = \begin{cases} \sum_{j \in C_{k,i}} \hat{f}_j(\theta_{L_{k-1}}) & \text{if not pruning} \\ R(f_i(\theta_{L_k}), x) + 2s^2p & \text{if pruning} \end{cases}$$

where $C_{k,i}$ is the set of all children nodes of clusterhead N_i at layer k . In this work, we are interested in studying the performance of a fixed-size sensor network in function of the threshold T_k . Thus, p is constant and the threshold $T_k(s)$ for level L_k , $k > 1$ is then

$$T_k(s) = 6s^2p + \sum_{j \in C_{k,i}} R(f_j(\theta_{L_{k-1}}), x) \quad (2)$$

III. OPTIMIZING PERFORMANCE IN A REAL SENSOR NETWORK

Although we do not take advantage of the self-locomotion capabilities of our robotic platform, we are ultimately interested in robotic sensor networks. Whereas in this paper, we consider as our energy costs the number of active (sensing) nodes in the network, our formalism can be extended to also take into account the additional cost of robot motion.

A. Problem Formulation

By implementing the threshold formula described above, theory in [18] states that the total partition size $|\theta|$ is reduced to an order of \sqrt{n} (in a network of a total n nodes). Under this assumption, and assuming the field's boundary type is known, the estimation reaches almost optimal bounds on MSE and network communication costs: The total energy used for communication (in- and out-of-network) is in the order of \sqrt{n} and the MSE decay rate is in the order of $1/n^\nu$, with $0 < \nu \leq 1$, where ν is defined by the boundary type. (For example, in the case where the boundary is described by a line, $\nu = 1/2$ and the MSE decay is in $O(1/\sqrt{n})$). Further, an optimal estimation of the field with (1) relies on i) a uniform distribution of nodes within the network, and ii) a measurement disturbance of zero-mean Gaussian noise, homogeneous for all sensors. With a decreasing MSE, the algorithm will balance the trade-off by systematically increasing the penalty associated with communication costs.

The approach in [18] optimizes the partitioning of the space, with a fixed number of active nodes by minimizing communication distances. Here, the problematic is different, because we try to optimize the number of active nodes by turning off all but one node per partition cell. Thus, no lower bounds (except the trivial bounds) on communication costs and MSE exist.

We introduce the objective function

$$f_{obj}(g(s), h(s)) \quad (3)$$

This function allows us to arbitrarily define the trade-off, with $g(s)$ the number of active nodes (i.e. energy consumed by the system) and $h(s)$, the resulting MSE. Both functions depend only on the threshold $T(s)$. Whereas in [19], s^2 is a fixed sensor noise variance, we here use it as a design parameter to modify the value of the threshold. The goal of this study is to define a methodology which finds the optimal value s that minimizes a user-defined cost function f_{obj} .

B. Expected Theoretical Performance

In the following we develop a formalism for the behavior of functions $g(s)$ and $h(s)$, which takes into account communication failure and a unique sensor model for each node. The idea of our formalism is fairly simple: at each level of the quadtree hierarchy, we predict the expected number of active nodes and the expected MSE, and propagate this value upwards to the next level. Thus, at the highest level in the hierarchy, we obtain the values which predict the performance of the whole network.

For all sensor nodes N_i in the network, we assume that the measured acoustic power x_i is sampled from a normal distribution with mean μ_i and variance σ_i^2 ($x_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$). We also assume that the network undergoes a communication failure rate of t_x . For reasons of simplicity, t_x is constant, but our approach is easily extended with arbitrary values $t_{i,j}$ for communication between a pair of nodes N_i and N_j . We are able to recursively calculate $g(s) = \mathbb{E}(A)$, the expected number of active nodes, and $h(s) = \mathbb{E}(MSE)$, the expected

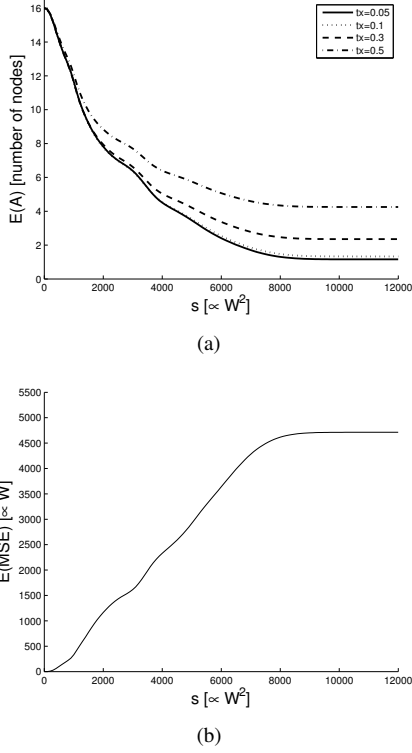


Fig. 5. Expected performance. (a) Total active nodes for varying transmission failure rates (b) MSE.

MSE. For any given layer k , for a clusterhead N_i we have

$$\begin{aligned} \mathbb{E}_{k,i}(A) &= (1 - P(U_{k,i})) \cdot \mathbb{E}_{k,i}(A|\bar{U}_{k,i}) \\ &\quad + P(U_{k,i}) \cdot \mathbb{E}_{k,i}(A|U_{k,i}) \end{aligned} \quad (4)$$

$$\begin{aligned} \mathbb{E}_{k,i}(MSE) &= (1 - P(U_{k,i})) \cdot \mathbb{E}_{k,i}(MSE|\bar{U}_{k,i}) \\ &\quad + P(U_{k,i}) \cdot \mathbb{E}_{k,i}(MSE|U_{k,i}) \end{aligned} \quad (5)$$

where $U_{k,i}$ represents the event that node N_i decides to prune its cluster. The probability $P(U_{k,i})$ that N_i prunes at layer k is

$$P(U_{k,i}) = P(u_{k,i}) \cdot \prod_{j \in C_{k,i}} P(U_{k-1,j})$$

because a clusterhead cannot prune its cluster if all its lower level clusters are not already pruned themselves, with

$$P(u_{k,i}) = P\left((R_{k,i} - \sum_{j \in C_{k,i}} R_{k-1,j}) < 6s^2p\right)$$

where $u_{k,i}$ is the event that node N_i decides to prune, knowing that its subclusters are already pruned. $C_{k,i}$ is the set of all children nodes of clusterhead N_i at layer k and $P_{0,i} = 1$ for all i , and

$$R_{k,i} = \sum_{j \in C_{k,i}} (f_j(\theta) - x_j)^2$$

where $f_j(\theta) = x_i$, and with $R_{0,i} = 0$ for all i . We note that our estimator $f_j(\theta)$ of node N_j , as described in section II-B.2, is the value x_i taken by its parent node N_i .

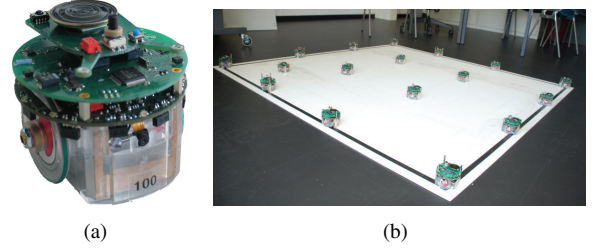


Fig. 6. The figure shows (a) the e-puck robot with the communication module stacked between the basic module and the jumper board (b) the real setup. 16 robots are evenly spaced out in a $1.5 \times 1.5 m^2$ large space. A 17th e-puck plays the role of a sound source

When we prune, we must take into consideration the failure of pruning messages. Thus, the expected number of nodes that can be deactivated upon pruning by node i at layer k is

$$\mathbb{E}_{k,i}(D) = \sum_{m=1}^{|C_{k,i}|-1} m (1 - t_x)^m t_x^{|C_{k,i}|-m-1} \quad (6)$$

The rate t_x does not affect the MSE because a clusterhead will not change its processing layer until it has received all measurement messages from its children. At layer L_0 we expect the values $\mathbb{E}_{0,i}(A) = 1$ and $\mathbb{E}_{0,i}(MSE) = 0$ for all i and thus we recursively obtain

$$\mathbb{E}_{k,i}(A|\bar{U}_{k,i}) = \sum_{j \in C_{k,i}} \mathbb{E}_{k-1,i}(A)$$

$$\mathbb{E}_{k,i}(A|U_{k,i}) = \sum_{j \in C_{k,i}} \mathbb{E}_{k-1,i}(A) - \mathbb{E}_{k,i}(D)$$

$$\mathbb{E}_{k,i}(MSE|\bar{U}_{k,i}) = \sqrt{\frac{1}{|C_{k,i}|} \sum_{j \in C_{k,i}} \mathbb{E}_{k-1,i}(MSE)^2}$$

$$\mathbb{E}_{k,i}(MSE|U_{k,i}) = \sqrt{\frac{1}{|F_{k,i}|} \sum_{j \in F_{k,i}} (x_i - x_j)^2}$$

where $F_{k,i}$ is the set of all nodes which descend from node N_i .

Figure 5 shows the predicted curves $g(s)$ and $h(s)$ for varying communication failure rates of 5%, 10%, 30% and 50%. Parameters μ_i and σ_i are extracted from field data of the experimental setup elaborated in the next section. We see that for an increasing threshold, the number of active nodes decreases, whereas the MSE increases.

C. Experimental Results

We perform our experiments on the e-puck robotic platform [5], [17]. The e-puck robot runs on a microcontroller of the dsPIC30 family. It has a trinaural microphone array which can sample sound (in parallel) at a maximal frequency of approximately 28.8 kHz. Also, it is equipped with a custom extension turret for short range communication which uses the subset of the 802.15.4 and ZigBee protocols present in TinyOS [4], [6]. The transmission power of the communication module is software controllable, and passes through

a custom attenuation circuit yielding effective maximum ranges between approximately 10cm and 5m—Figure 3.7 in [6] shows this behavior. Here, we set a maximal transmission power. Fig. 6 (b) shows the setup with 16 robotic nodes in a $1.5 \times 1.5 \text{ m}^2$ arena. An additional robot plays the role of a stationary sound source. The source robot generates a continuous acoustic field by emitting white noise at a constant intensity, and is randomly placed in the arena at the beginning of each experimental run. Each sound measurement (acoustic power) is computed from the raw sound intensities collected by the microphone array.

1) Simulation Results: First, we validate our network topology and robot controller. We perform a preliminary experiment where we feed real field data into the robotic simulation platform Webots [16], simulating only radio communication. The radio communication is realistically modeled within the simulation software using a plugin based on OMNeT++ [21], which accurately simulates the physical layer (i.e., with channel fading) and data link layer (i.e. modulation properties, channel coding, MAC protocol).

Figure 7 summarizes the behavior of the two control variants NS and TBS as elaborated above, with respect to (a) the number of active nodes and (b) the MSE. We performed 500 runs per threshold, for 24 different thresholds with s in $[0..12000]$. For NS, the total number of active nodes as well as the resulting MSE will remain constant. As expected for

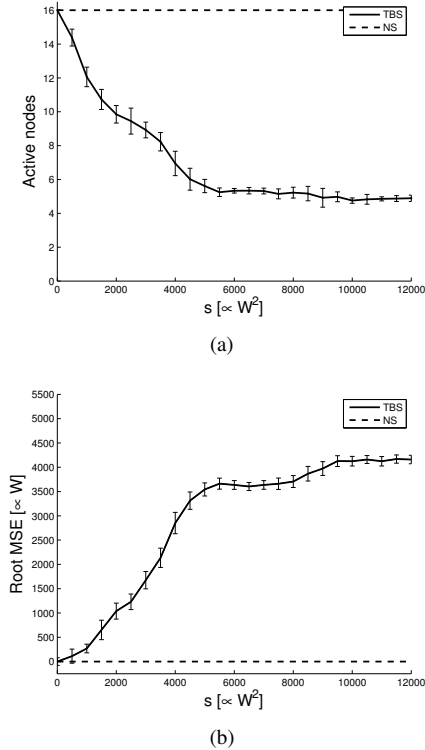


Fig. 7. Performance with i) NS and ii) TBS. 500 runs were performed per threshold, for 24 different thresholds with s in $[0..12000]$. (a) Total active nodes (b) MSE. The errorbars show a 95% confidence interval.

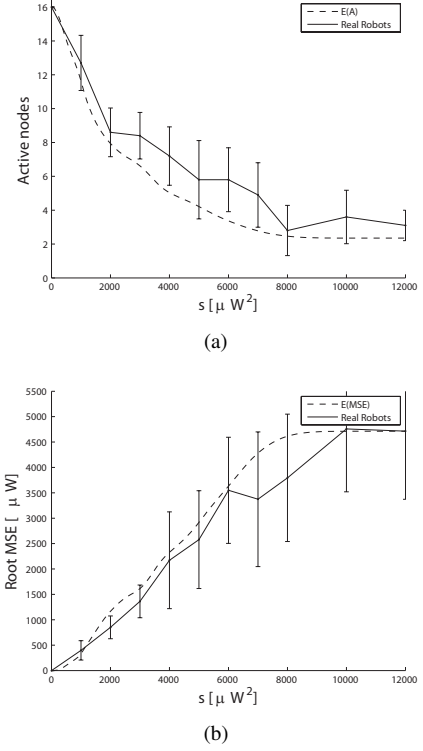


Fig. 8. Real robot results as well as analytic results for $t_x = 0.3$. 10 runs were performed per threshold, for 12 different thresholds with σ in $[0..12000]$. (a) Total active nodes (b) MSE. The errorbars show a 95% confidence interval.

TBS, we observe a decreasing number of active nodes and an increasing MSE as the threshold increases.

2) Real Robot Results: Figure 8 shows results obtained on the real setup. We performed 10 runs per threshold, for 12 different thresholds with s in $[0..12000]$. The values show good correspondence with the results obtained through simulation (in Fig. 7). Further, we superimpose the analytical curves $\mathbb{E}(A)$ and $\mathbb{E}(MSE)$ for a transmission failure rate of $t_x = 0.3$. This value corresponds to the failure rate measured in this experiment. The analytical curves show good correspondence to the results obtained with the physical testbed.

D. Optimization of f_{obj}

We have seen above that the theoretical analysis provides a good approximation of the system's behavior. After an initial observation phase performed by the sensor network (where all nodes are active), sensor models σ_i, μ_i are extracted and disseminated. Thus, the network can use the prediction formalism to estimate optimal thresholds, according to a specific setup and user needs. Here, we demonstrate a potential user case-study by applying our methodology on an arbitrary objective function. We define

$$f_{obj} = \alpha g(s) + \beta h(s)$$

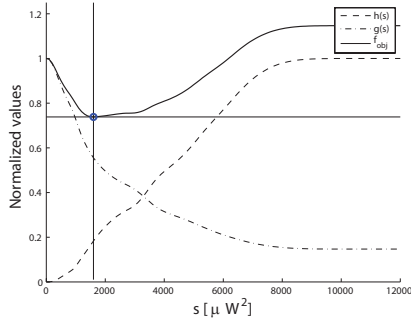


Fig. 9. f_{obj} for $\alpha, \beta = 1$ and $t_x = 0.3$. The predicted optimal value is $s = \operatorname{argmin}_{s \in \mathbb{R}^+} f_{obj}(s) \approx 1600$

with $\alpha, \beta = 1$. For a communication failure rate of $t_x = 0.3$, and using the sensor models σ_i, μ_i extracted from our experimental field data, we have an optimal value for $s = \operatorname{argmin}_{s \in \mathbb{R}^+} f_{obj}(s) \approx 1600$. This result is shown in Fig. 9.

IV. CONCLUSION & FURTHER WORK

In this work we first develop a layer-based fully asynchronous distributed node controller, specific to hierarchical network topologies, where it performs self-configuration based on an estimation technique. Whereas the theory for the estimation technique optimizes communication costs, we decouple our performance metric by considering a sensor-node as either fully active or shut-down. Because a real sensor network violates the uniform Gaussian noise assumption, its performance is not easy to predict. By varying the pruning threshold as an algorithmic parameter, we analyzed the system's performance in realistic simulation as well as on hardware, and developed a formalism that accurately captures the behavior of a real sensor network. Finally, we develop a framework which ultimately allows for a specific, user-defined trade-off between the cost and accuracy of a sensor network.

There are a number of possible extensions to this work. First, an augmented node control should envision the reactivation of nodes absorbed by the idle state. This is especially interesting for sensor networks deployed in non-static, dynamic environments. Also, the introduction of clusterhead rotation cycles and distributed node responsibilities lead to increased robustness, which is a key factor for large-scale networks, composed of potentially unreliable and heterogeneous nodes.

REFERENCES

- [1] Tarik Arici and Yucel Altunbasak. Adaptive sensing for environment monitoring using wireless sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2347–2352, Atlanta, GA, USA, March 2004.
- [2] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 85–96, New York, NY, USA, 2001. ACM.
- [3] Chee Yee Chong and S. P. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, August 2003.
- [4] Christopher M. Ciani, Jim Pugh, and Alcherio Martinoli. Exploration of an incremental suite of microscopic models for acoustic event monitoring using a robotic sensor network. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3290–3295, Pasadena, CA, USA, May 2008. IEEE Press.
- [5] Christopher M. Ciani, Xavier Raemy, Jim Pugh, and Alcherio Martinoli. Communication in a swarm of miniature robots: The e-puck as an educational tool for swarm robotics. In *Simulation of Adaptive Behavior (SAB-2006), Swarm Robotics Workshop*, pages 103–115, Rome, Italy, October 2006, Lecture Notes in Computer Science (2007), vol. 4433.
- [6] Christopher Michael Ciani. *Distributed intelligent algorithms for robotic sensor networks monitoring discontinuous anisotropic environmental fields*. PhD thesis, Lausanne, 2009.
- [7] David Culler, Deborah Estrin, and Mani Srivastava. Overview of sensor networks. *IEEE Computer, Special Issue in Sensor Networks*, 37(8):41–49, August 2004.
- [8] M. Demirbas and Xuming Lu. Distributed quad-tree for spatial querying in wireless sensor networks. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 3325–3332, June 2007.
- [9] T. Dubitzki, A. Y. Wu, and A. Rosenfeld. Parallel region property computation by active quadtree networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(6):626–633, 1981.
- [10] R. A. Finkel and J. L. Bentley. Quad trees: A data structure for retrieval on composite keys. *Journal Acta Informatica*, (1):1–9, 1974.
- [11] Antonin Guttman. R-trees: a dynamic index structure for spatial searching. *SIGMOD Rec.*, 14(2):47–57, 1984.
- [12] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 10 pp. vol.2–, Jan. 2000.
- [13] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 56–67, New York, NY, USA, 2000. ACM.
- [14] S. Sitharama Iyengar and Richard R. Brooks. *Distributed Sensor Networks*. Chapman & Hall/CRC, USA, 2005.
- [15] Juan Liu, James Reich, and Feng Zhao. Collaborative in-network processing for target tracking. *EURASIP Journal on Applied Signal Processing*, (4), 2003.
- [16] Olivier Michel. Webots: Professional mobile robot simulation. *Journal of Advanced Robotic Systems*, 1(1):39–42, 2004.
- [17] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Ciani, A. Klapotcz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli. The e-puck, a robot designed for education in engineering. In *Proc. of the 9th Conference on Autonomous Robot Systems and Competitions*, pages 59–65, Castelo Branco, Portugal, May 2009.
- [18] Robert Nowak and Urbashi Mitra. Boundary estimation in sensor networks: Theory and methods. In *Information Processing in Sensor Networks (IPSN)*, pages 85–90, Palo Alto, CA, USA, April 2003. Springer-Verlag.
- [19] Robert Nowak, Urbashi Mitra, and Rebecca Willett. Estimating inhomogeneous fields using wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 22(6):999–1006, August 2004.
- [20] Hanan Samet. The quadtree and related hierarchical data structures. *ACM Comput. Surv.*, 16(2):187–260, 1984.
- [21] Andras Varga. Software tools for networking: “OMNeT++”. *IEEE Network Interactive* (2002), 16(4), July 2002.
- [22] Rebecca Willett, Aline Martin, and Robert Nowak. Backcasting: Adaptive sampling for sensor networks. In *Information Processing in Sensor Networks (IPSN)*, pages 124–133, Berkeley, CA, USA, April 2004. ACM Press.
- [23] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 70–84, New York, NY, USA, 2001. ACM.
- [24] Feng Zhao, Jie Liu, Juan Liu, L. Guibas, and J. Reich. Collaborative signal and information processing: an information-directed approach. *Proceedings of the IEEE*, 91(8):1199–1209, Aug. 2003.